AN INNOVATIVE APPROACH FOR TRAINING IN PROGRAMMING USING VIRTUAL LABORATORIES

Galya Shivacheva, Veselina Nedeva, Venelin Bochev, Emilia Dimova
Trakia University of Stara Zagora,
Faculty of Technics and Technologies of Yambol
38 Graf Ignatiev Str., Yambol, Bulgaria
phone: +359 46 669181, e-mail: shivacheva_g@abv.bg

Abstract: The paper describes and analyses the computer programming educational technology in Faculty of Technics and Technologies of Yambol, using traditional methodologies, e-learning and virtual laboratories. It reviews the characteristics of computer programming education, focusing on animation and simulations created by other higher education institutions on algorithm visualizations. The innovative approach has connected with using Moodle Virtual Lab and the Open Judge System for automatic students' assessment and testing. In addition, the Virtual Laboratory for Training in Programming (VLTP) provides a virtual environment with animations, simulations and visualizations of the basic terms, algorithms and data structures in programming sphere.

Keywords: Virtual Laboratory, training in programming, Moodle Virtual Programming Lab, Open Judge System, automatic testing, self-testing programming source code.

1. INTRODUCTION

Training in programming includes absorption of theoretical knowledge and achievement of practical skills for understanding and using of algorithms and data structures in a specific programming language and creating a programming code. Theoretical knowledge is related to the basic concepts in the programming area, most of which are abstract and their assimilation requires overcoming some difficulties and the need for understanding and comprehension by students. Practical skills for describing and application of algorithms require development of a new type of thinking – algorithmic. It allows describing the sequence of actions that will be implemented by a computer and therefore need to be extremely precise, clear and formal.

In the training on programming in Informatics course, the most challenging for students is self-test (debugging and testing their own programs). Further validation of the performance of educational tasks by teacher is the most time-consuming part of the learning process and it involves the elaboration of methods of estimation of expected future solutions. If the test solving theoretical problems focused on the analysis of solutions and corrects the course of the answers, the situation is different in solving practical problems. Test examination of the curriculum program aims to verify the correctness of the working program in all supposed practical situations specified in the condition of the task and to evaluate their effectiveness.

The verification of this program without the direct student participation has developed only if the task is clear and full. We pay particular attention to the description of the formats of input and output data, the type of input parameters and output in the condition of tasks. The inclusion of test examples in input and output data in the task condition is useful. This allows the student to check if he properly understand the task condition [6].
2. METHODS AND TECHNOLOGY OF TRAINING IN PROGRAMMING AT FACULTY OF TECHNICS AND TECHNOLOGIES OF YAMBOL

We applied mixed form of education at Faculty of Technics and Technologies of Yambol. It is a combination of traditional, e-learning and virtual laboratory. For E-learning lecturers use virtual learning environment MOODLE, which has personalized for the needs of Trakia University - e-TrUni [5]. In e-TrUni there is developed C++ programming course, which offers students: syllabus curriculum, training and assessment technologies, vocabulary, videos, lectures, presentations, practical exercises designed to methodological guidance for use. In this environment, the training of students carries out as follows:

Students attend lectures to learn the theoretical material in C++ programming or prepare the material independently using textbooks and presentations offered in the e-learning course "Programming and Computer use - Part II." The theoretical material has presented with demonstration and explanation of the source code of the sample programs. The students have the opportunity to participate directly experimenting with examples of their computers, to ask questions, to discuss and to debate.

The participation in the practical exercises is mandatory. The students developed programs on already assimilated theoretical material in them. Upon completion of this part of the training, they must have learned and acquired practical skills for implementation of data types, operators, structures and basic algorithmic structures – chain, branching, cycle, sets and sets operations, symbol data and operations with them. In e-TrUni there are developed themes of all practical exercises with task conditions, block diagram algorithm for each of them and questions at the end of each exercise. This can be useful for the students who require additional training for self-study, for midterm or the final exam.

Each thematic unit has additional conditions of tasks for self-study, which are suitable for improving skills in programming. For the more difficult topics in C++ programming in the course are integrated video tutorials. There are indicated resources of Internet, which can be useful for personal independent learning.

Teaching technology in C++ programming includes Course work development which each student has to prepare and to attach his file in e-TrUni. At the end of the semester, the students have to defend his Course work. During the semester, students can do a test on the training material. The test aims to examine depth knowledge of the material and contains questions of different types, incl. fragments of programs, for which the lecturer expects introduction of the results of their performance by the keyboard. The similar test students have to do during the final exam.

They received also a ticket with three tasks with different levels of difficulty. After development tasks with C++ programs they save the source code of the program and a screenshot with the results of the implementation of the program. After the presentation of their programs, the code has discussed with the lecturer and he asks specific questions about its implementation. The evaluation has based on the average score of test results, source code of the tasks and the Course work. A certain number of bonus points bring them active participation in the lectures, exercises, and shown results during the semester.

The existing training system combines the traditional approach and uses the advantages offered by e-learning. E-TrUni allows identifying and administering forums, leaving messages and run videoconferencing communications. The disadvantage is the lack of opportunity to check the source code during the self-training students. To overcome this disadvantage in learning we apply innovative approach using virtual labs: Virtual Lab in Moodle and Open Judge System.
3. DISCUSSION AND RESULTS

3.1. Specific features of the training in programming

At introduction to the main concepts in the C++ language program for better understanding of the material, the teacher’s explanation is supported with practical examples that can be presented graphically in a presentation or with animation. The term “variable” may be presented as a box with a label (“variable” name) and content - (“variable” value).

The absorption of operators for conditional statement - Conditional operator and multiple choice may use visualization block diagram to solve the task by checking the condition from the electronic textbook [7]. The student is able to introduce a variable to see which branch of the algorithm has executed.

The students understand the loop operators if we focus on examples and visualize the process through block diagram illustrating the implementation of the cycle gradually when put into reverse order the digits of one number [8]. In testing of loop operator with „managing variable“ it is correct through every step to derive the value of the „managing variable“ and the values of other variables in the cycle body. Thus, the students better understand this operator and its operation.

To justify the need for a new data structure as the array for processing the sequence of numbers, which to date has been done with a simple variable, the lecturer should place task in solving that students need examine the row more than once. For example, to exchange a minimum and maximum number in a sequence of numbers. In this situation, the lecturer can use animations for elements of the array, and the appropriation of their values. For the detection of differences between the index and the value of the elements, it is desirable in animation to stress on the index and displays of its name and its value [9].

In introducing to functions and recursion at the beginning, the lecturer has to made justification for the need to use functions. A good example is the task of finding the largest of three numbers as the lecturer twice uses the function of finding the larger of the two numbers. Suitable animation for acquiring the knowledge is the animation of David Galles [11] from the University of San Francisco. He creates it with canvas element of HTML5 and programming code written in JavaScript interactive animations for visualization of three recursive subprograms - to calculate n!, reversing a string and the task for n queens (The task is to arrange N queens on the chessboard NxN in a manner that no two of them can not endanger each other). The basic algorithms can be presented with short animations that displayed a similar example of the practice - for example, the algorithm for changing the values of two variables with the changing of content of the two glasses using a third empty glass (i.e. the third additional variable). The students perceive much more easily the algorithm for minimum or maximum number in a range of numbers by displaying in the virtual laboratory through animation, video or as the other algorithms, connected to the elements of an array, presented schematically as rectangles with different heights and comparing with the current minimum or maximum element. The lecturer visualizes the algorithm for consistently search an element in an array in VLOP in this manner: he marks by different colors: examined elements (yellow), unexamined elements (green), found elements (red) and searching elements (cyan) - (Figure 1).

In sorting bubble algorithm, we can show the values of the elements that we exchange or by the algorithm for sorting through selection to take the next minimum or maximum element, depending on which algorithm option we implement.
In the training on programming, we use the experience of other universities. For example, at the University of Komarno – Slovakia, they have created a rich collection of interactive animations for teaching algorithms and programming. It is available on the Internet at the addresses specified in [1] and [2]. These animations and visualizations represent clearly step by step each of the algorithms. They help students to percept easier lectures. Another advantage is the possibility to repeat many times, and it can be at convenient time and place for the user. The only disadvantage is that there is not possibility the user to enter his input data and to experiment with it, and users work with the same set of animations’ authors. For visualization of sorting algorithms, we can use interactive animations of David Galles from the University of San Francisco. [11]

3.2. Innovative approach in training in programming

3.2.1. Virtual Lab in Moodle

There is integrated module Virtual Programming Lab (VPL) in e-TrUni to overcome the disadvantages. It checks the tasks’ code whose conditions lecturers specify in advance. Moodle VPL- Virtual Programming Lab is an activity module that can develop evaluation of program code and has following main features [10]:

Element with value 30 is with index 1.
- It allows editing the source code of the programs in the browser using the applet;
- The students can run interactively their programs in the browser;
- Students can run tests to check their programs;
- It allows searching of similarity program code between files of other students;
- It allows limiting editing functions and avoiding copying and insertion of external text as a code for the program.

This module has realized as plug-in of virtual learning environment Moodle. It is responsible for input, control and compilation of program code. It can work with Ada, C, C++, C#, Fortran, Haskell, Java, Octave, Pascal, Perl, PHP, Prolog, Python, Ruby, Scheme, SQL and VHDL. In FTT – Yambol we use C++ and Java.

The sequence of student’s actions for a program execution and testing present in following steps: [10]

- There is prepared file with student's program.
- The lecturer also has prepared the file "Execution files", which has the same name as the student's name.
- Depending on actions (run, debug or evaluate) the lecturer prepares the script file. If there is no script VLP takes script by default that automatically recognizes the programming language based on the extension of the saved file;
- In case of assessing if there is no script, VLP adds evaluation program. The basis of the program is on input and output and requires specifying cases for testing in file "vpl_evaluate.cases".
- The student send accumulated files to the server for execution.
- The module VPL informs browser that the implementation has already begun.
- If the requirement is to evaluate when you complete the task, the result of the evaluation has extracted from the server for execution. The evaluation of VLP, together with the code of the program are in Figure 2.

![Figure 2. Source code of the program and test results.](image-url)
E-TrUni integrated module Virtual programming lab includes the following topics in two groups: Basic CPP (Data types, Variable types and scope, Constants; Operators, Basic Inputs-Outputs; Conditional statement; Loop types: for, while, do-while; Functions, Numbers, Strings; Arrays; Pointer, References;) and Data Structures and Algorithms (Stacks, Queues, and Recursion; Vectors, Lists, and Recursion; Trees; Priority Queues; Search Trees; Sorting and Selection; Graphs).

After saving of program in VPL the students can use next functionalities in the menu: Description, Submissions list, Similarity, Test activity, Submission, Edit, Submission view, Grade, Previous submissions list (Figure 3).

3.2.2. Open Judge System.

Further development of the idea of innovative training in programming has achieved by using Open Judge System. It has developed by Nikolay Kostov and implemented in www.bgcoder.com. There are the experiments in it. We will install it soon on a remote server and it will be launch from the virtual learning environment of e-TrUni.

OJS (Open Judge System) is a system with open code for online testing competitions for algorithm work. The lecturer uses it on competitive principle to test the knowledge of programming on five programming languages: C#, C++, JavaScript, Java, Python and PHP. The system consists of few main elements [12]:

- Web based interface for users and administrators (ASP.NET MVC);
- Windows service for compiling, executing and checking the sent by the user files;
- Restrictions in the execution of the programs realized by Windows APIs;
- Checks the source code and compile the file [12].
The checking has realized with specially developed software scripts Checkers [3], which are the following types:

- **Exact** - the system result of the sent code result with the expected output symbol by symbol;
- **Trim** - the system removes the empty spaces before and after the result of sent code and then compares symbol by symbol with the expected output;
- **Sort** - the system sorts all lines in the sent code result and then compares symbol by symbol with the expected output;
- **Case-insensitive** - the system does not make difference between small and capital letters when comparing the sent code result with the expected output symbol by symbol;
- **Precision N** - the accuracy has checked for tasks with the result a decimal number.

BgCoder is a competitive system with open code created and maintained by the authors of the Open Judge System. It performs automated assessment of each sent by students source code. The system compiles the code, compares the outcome of its implementation on a preset input data, and compares the resulting output with an expected result.

For this purpose, we have created previously input and output tests. The users submit input tests to the test program, and the output tests have compared as strings with the result of the implementation of student code. In the statement of the task, the lecturer must give the exact format of output and at least one sample of input and output. The lecturer assigns in the system maximum operating time of the algorithm and limitations on the amount of RAM. A number of tests on an algorithm carries out the checking of the tasks. The lecturer adjudges relevant points based on the results. When the students upload the task the system, give them one of the following results:

- ✔ = Correct answer.
- ✗ = Wrong answer.
- ☻ = Time limit – Outstanding is the maximum operating time of the algorithm.
- ⚠ = Run-time error – The program interrupts the execution of work and throws an exception (Error performance).

4. CONCLUSION

Virtual laboratory for training in programming is a not just a substitute for lectures and practical exercises in a computer room. It provides a virtual environment with animations, simulations and visualizations. It can present the implementation of the algorithm gradually and not just as sequence of numbers and text, but visually with the help of geometric shapes (e.g. rectangles of varying heights, proportional to the value of the number they figure). Implementation of innovative approach to training allows achieving visibility, lower level of abstraction during the learning process, the possibility of development of algorithmic thinking, independent verifying and checking of source code by the student and an immediate assessment of the proposed for assessment program. This approach of application of virtual lab creates the right conditions for more targeted independent study outside classes. It saves time for the lecturer to check the source code and allows students to assess the achieved level of preparation. It allows stimulating the students’ work and limits the number of attempts to compile the source code.
5. RESULTS

Because of the experience, underlying observations, studies and experiments in education in programming C++ we can make the following conclusions:

1. The application of new approaches to teaching and learning programming allows fuller absorption of the material and its understanding in depth.
2. Virtual labs and websites programming, created for training and learning have stimulating effect to enhance the students' knowledge.
3. The integrated module Virtual Lab in Moodle in the virtual learning environment of Trakia University - e-TrUni develops better conditions for self-study with functionalities for entering, editing, compilation and evaluation of the created source code.
4. Using Open Judge System introduces the competition principle in the preparation and presentation of students, which affects the increasing of their interest in learning programming and allow to check the created source code and to assess the achievements of the task.

The innovative approach will be fully apply in the Virtual laboratory for training in programming (VLTP). Any topic from the curriculum has presented in it by: theoretical material; source code; visualization, animation and simulation; opportunity for compilation and verification of such a task by testing; assessment the progress of the test result compilation.

We continue research the effects of the virtual laboratory approach on success and acquired practical students' skills.

6. ACKNOWLEDGMENT

This paper is a contribution to the implementation research project No2-FTT/30.04.2015 “Application of the virtual laboratories at the universities” of the Faculty of Technics and Technologies of Yambol, Trakia University (Bulgaria).

7. REFERENCES